LAB4: LOGIC, SHIFT & ROTATE INSTRUCTIONS

Done by: Abeer(www.UOB-BH.com)

Objective:

- Implement logic instructions

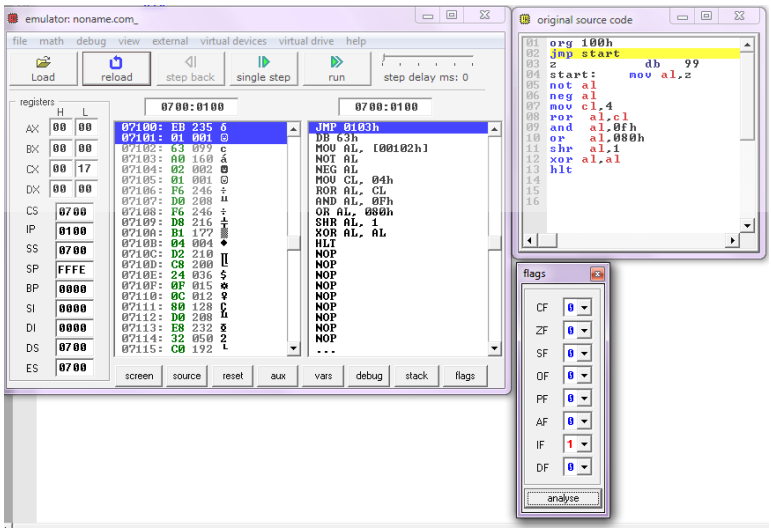
Procedure:

Run emu8086 program

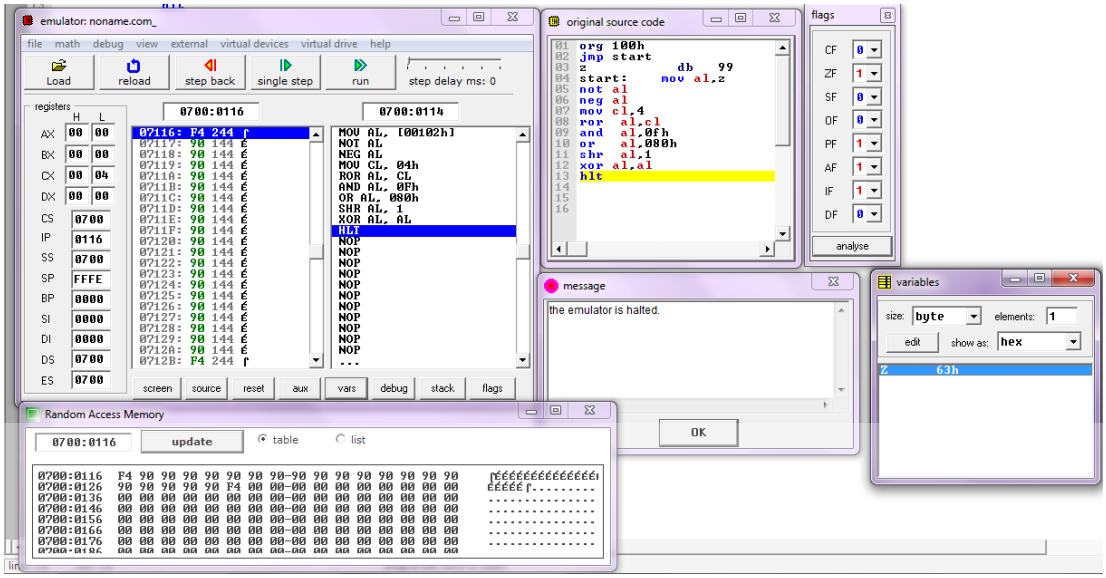
1. **In the welcome window click new button**
2. **In the choose template window select empty workspace**
3. **In the Source Code Editor window type the following (tiny) program.**

```
        org 100h
        jmp start
z       db  99
start:  mov al,z
        not al
        neg al
        mov cl,4
        ror  al,cl
        and  al,0fh
        or   al,080h
        shr  al,1
        xor al,al
        hlt
```

4. **Simulate the program by clicking on emulate button.**
5. **To view the effect on the processor flags during execution, open the flag window by clicking on the flag button, the flag window will appear.**
6. **Move the flags window to a proper location so that you can view the flags simultaneously with the main emulator window.**
7. **View and record the contents of all registers and flags.**



8. **Press on the *step* button to execute the program step by step, and view carefully the affected registers and flags. Record them all.**



9. Try the following program which multiplies z by 14 and saves product in ax

```
            org 100h
            jmp start
z           db  75
start:      mov al,z
            mov ch,14
            mul ch
            hlt
```

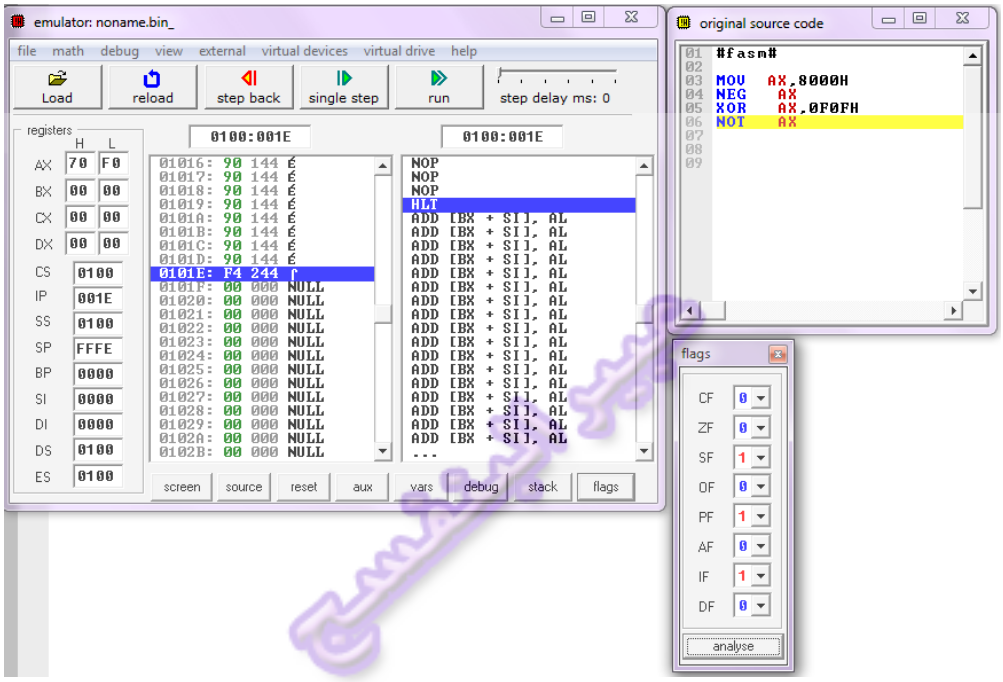
10. Home Work: Without using mul instruction, write and implement a program to multiply z by 14 and save product in ax. Hint: use shift instruction.
    Compare the execution speed and size of your program with that of step 9.

```
Mov ax,z
SHL ax,1
mov cx,ax
SHL ax,1
mov bx,ax
SHL ax,1
add ax,cx
add ax,bx
hlt

z db 66
```

My program is faster than the program in step 8, but the size of the first program is smaller than my program

11. **Implement and discuss how the flags and registers are affected by the following instructions:**
    MOV AX,8000H
    NEG AX
    XOR AX,0F0FH
    NOT AX



Report

a. **Write the program lines by which the following logical function may be implemented:**

$$F = \overline{\left[\left(A + \overline{B.C}\right) \oplus \overline{\left(A + B + C\right)} \right] . \; A \; . \overline{C}} \; + D$$

Where A, B, C, D are 16-bit numbers in AX, BX, CX, DX respectively

b. **Show and discuss all your results.**

JMP x

```
A dw ----------
B dw ----------
C dw ----------
D dw ----------
F dw ?

X:  mov SI,ax
      Mov DI,bx
     AND bx,cx
     NOT bx
     OR bx,ax
    OR ax,DI
    OR ax,cx
    NOT ax
     XOR ax,bx
    AND ax,SI
   NOT cx
  AND  ax,cx
  NOT  ax
  OR ax,Dx
 MOV F,ax
hlt
```